

TITLE: METHOD AND APPARATUS FOR DISTRIBUTING
DIGITIZED STREAMING VIDEO OVER A NETWORK

INVENTOR: DAVID A. MONROE, RAYMOND R. METZGER

BACKGROUND OF INVENTION:

1. FIELD OF INVENTION:

[0001] The invention is generally related to digital video transmission systems and is specifically directed to a method and apparatus for compressing and distributing digitized video over a network for supporting the transmission of live, near real-time video data.

2. DESCRIPTION OF THE PRIOR ART:

[0002] Cameras employ digital encoders that produce industry-standard digital video streams such as, by way of example, MPEG-1 streams. The use of MPEG-1 streams is advantageous due to the low cost of the encoder hardware, and to the ubiquity of software MPEG-1 players. However, difficulties arise from the fact that the MPEG-1 format was designed primarily to support playback of recorded video from a video CD, rather than to support streaming of 'live' sources such as surveillance cameras and the like.

[0003] MPEG system streams contain multiplexed elementary bit streams containing compressed video and audio. Since the retrieval of video and audio data from the storage medium (or network) tends to be temporally discontinuous, it is necessary to embed certain timing information in the respective video and audio elementary streams. In the MPEG-1 standard, these consist of Presentation Timestamps (PTS) and, optionally, Decoding Timestamps (DTS).

[0004] On desktop computers, it is common practice to play MPEG-1 video and audio using a commercially available software package, such as, by way of example, the Microsoft Windows Media Player. This software program may be run as a standalone application. Otherwise, components of the player may be embedded within other software applications.

[0005] Media Player, like MPEG-1 itself, is inherently file-oriented and does support playback of continuous sources such as cameras via a network. Before Media Player begins to play back a received video file, it must first be informed of certain parameters including file name and file length. This is incompatible with the concept of a continuous streaming source, which may not have a filename and which has no definable file length.

[0006] Moreover, the time stamping mechanism used by Media Player is fundamentally incompatible with the time stamping scheme standardized by the MPEG-1 standard. MPEG-1

calls out a time stamping mechanism which is based on a continuously incrementing 94 kHz clock located within the encoder. Moreover, the MPEG-1 standard assumes no Beginning-of-File marker, since it is intended to produce a continuous stream.

[0007] Media Player, on the other hand, accomplishes time stamping by counting 100's of nanoseconds since the beginning of the current file.

SUMMARY OF INVENTION

[0008] The video system of the subject invention is adapted for supporting the use of a local-area-network (LAN) or wide-area-network (WAN), or a combination thereof, for distributing digitized camera video on a real-time or "near" real-time basis. Certain algorithms or methods used in the camera encoders and in the display stations are disclosed and form the nexus of the invention.

[0009] The subject invention is specifically directed to a method for recognizing and playing a continuous streaming video data signal with no known beginning of data signal and no known end of data signal, by assigning an arbitrary beginning of data signal to the streaming video in mid-stream, and assigning an arbitrary end of data signal to the streaming video for identifying the length of the video stream. The continuous streaming video may be time stamped. In the described embodiment the beginning of data signal is assigned by arbitrarily assigning a zero value to the first time stamp received. The end of data signal is arbitrarily set at a number sufficiently high to accommodate the functional life of the system based on the capability of the player platform utilized. In the preferred embodiment, the end of data signal is set at the highest number achievable by the player platform.

[0010] In the preferred embodiment of the invention, the system uses a plurality of video cameras, disposed around a facility to view scenes of interest. Each camera captures the desired scene, digitizes the resulting video signal, compresses the digitized video signal, and sends the resulting compressed digital video stream to a multicast address. One or more display stations may thereupon view the captured video via the intervening network.

[0011] In an exemplary embodiment, a common MPEG-1 encoder is used to perform the actual MPEG compression of a digitized camera signal. An example encoder is a W99200F IC, produced by Winbond Corporation of Taiwan. This IC produces an MPEG Video Elementary Stream that contains the appropriate PTS information as mandated by the MPEG standard. A proprietary algorithm converts the MPEG PTS data into the format required by the Microsoft

Media Player.

[0012] When invoking Media Player to view the streaming camera video, it is first necessary to inform Media Player of the file length since the camera produces a stream rather than a discrete file, the file length is undefined. In the exemplary embodiment, the Media Player's 63-bit file length variables are all set to 1. Media Player compares this value to a free-running counter that counts ticks of a 10 MHz clock. This counter is normally initialized to zero at the beginning of the file. Given 63 bits, this permits a maximum file length of approximately thirty thousand years. This effectively allows the system to play streaming sources.

[0013] A problem with this approach arises when additional users attempt to connect to a stream that is already in progress. Media Player expects that file length and other related information is normally sent only once, in a file header, and is not periodically repeated. Thus, users connecting later will not receive the file length information contained in the header. This problem is resolved by developing a software 'front-end' filter that examines and modifies data being passed from the network to Media Player. This software formulates a dummy video file header, and passes it to Media Player. The filter then examines the incoming video stream, finds the next sequential Video Header, and thereupon begins passing the networked video data to the Media Player decoder and renderer. This effectively allows users to 'tune in late', by providing Media Player with an appropriate file header.

[0014] A further issue arises when the networked video data is passed to Media Player. Since the user has connected to the video stream after the start of the file, the first timestamp received by Media Player will be non-zero, which causes an error. To overcome this problem, the novel front-end software filter replaces each received timestamp with a value calculated as the current timestamp minus first timestamp received. This effectively re-numbers the timestamp in the local video stream starting with an initial value of zero.

[0015] The subject invention permits any given source of encoded video to be viewed by more than one user. While this could hypothetically be accomplished by sending each recipient a unique copy of the video stream, such an approach is tremendously wasteful of network bandwidth. The subject invention resolves this by transmitting one copy of the stream to multiple recipients, via Multicast Routing. This approach is commonly used on the Internet, and is the subject of various Internet Standards (RFC's). In essence, a video source sends its video stream to a Multicast Group Address, which exists as a port on a Multicast-Enabled network router or

switch. It will be understood by those skilled in the art that the terms "router and/or switch" as used herein is intended as a generic term for receiving and rerouting a plurality of signals. Hubs, switched hubs and intelligent routers are all included in the terms "router and/or switch" as used herein. The router or switch then forwards the stream only to IP addresses having known recipients. Furthermore, if the router or switch can determine that multiple recipients are located on one specific network path or path segment, the router or switch sends only one copy of the stream to that path. From a client's point of view, the client need only connect to a particular Multicast Group Address to receive the stream.

[0016] At present there is not a standardized mechanism for dynamically assigning these Multicast Group Addresses in a way that is known to be globally unique. This differs from the ordinary Class A, B, or C IP address classes. In these classes, a regulatory agency assigns groups of IP addresses to organizations upon request, and guarantees that these addresses are globally unique. Once assigned this group of IP addresses, a network administrator may allocate these addresses to individual hosts, either statically or dynamically using DHCP or equivalent network protocols. This is not true of Multicast Group Addresses; they are not assigned by any centralized body and their usage is therefore not guaranteed to be globally unique. Thus, in accordance with the subject invention as presently configured, each video encoder must possess two unique IP addresses – the unique Multicast Address used by the encoder to transmit the video stream, and the ordinary Class A, B, or C address used for more mundane purposes. Therefore, it is necessary to provide a means to associate the two addresses, for any given encoder.

[0017] Pending the release of improved Internet Group Multicast Protocols, The subject invention provides a mechanism for associating the two addresses. This method establishes a sequential transaction between the requesting client and the desired encoder.

[0018] First, the client requesting the video stream identifies the IP address of the desired encoder. Once the encoder's IP address is known, the client obtains a small file from the desired encoder, using FTP, TFTP or other appropriate file transfer protocol over TCP/IP. The file, as received by the requesting client, contains various operating parameters of the encoder including frame rate, UDP bitrate, image size, and most importantly, the Multicast Group Address associated with the encoder's IP address. The client then launches an instance of Media Player, initializes the front-end filter, and directs Media Player to receive the desired video stream from the defined Multicast Group Address.

[0019] Streaming video produced by the various encoders is transported over a generic IP network to one or more users. User workstations contain one or more ordinary PC's, each with an associated video monitor. The user interface is provided by an HTML application within an industry-standard browser, for example, Microsoft Internet Explorer.

[0020] Streaming video signals tend to be bandwidth-intensive. To address this, each encoder is equipped with at least two MPEG-1 encoders. When the encoder is initialized, these two encoders are programmed to encode the same camera source into two distinct streams: one low-resolution low-bitrate stream, and one higher-resolution, higher-bitrate stream.

[0021] It is, therefore, an object and feature of the subject invention to provide the means and method for displaying "live" streaming video over a commercially available media player system.

[0022] It is a further object and feature of the subject invention to provide the means and method for permitting multiple users to access and view the live streaming video at different time, while in process without interrupting the transmission.

[0023] It is a further object and feature of the subject invention to permit conservation of bandwidth by incorporating a multiple resolution scheme permitting resolution to be selected dependent upon image size and use of still versus streaming images.

[0024] It is an additional object and feature of the subject invention to provide for a means and method for identifying an artificial file length for a continuous streaming video.

[0025] It is also an object and feature of the subject invention to provide a means and method for artificially identifying a "beginning-of-file" signal for a continuous streaming video.

[0026] It is a further object and feature of the subject invention to provide for a means and method for combining an IP address in accordance with accepted nomenclature with an encoder address to provide a unique global address for each encoder associated with a streaming "live" video system.

[0027] Other objects and feature of the subject invention will be readily apparent from the accompanying drawings and detailed description of the preferred embodiment.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] Fig. 1 is a block diagram of a typical multi-camera system in accordance with the subject invention.

[0029] Fig. 2 is an illustration of the scheme for multicast address resolution.

[0030] Fig. 3 illustrates a typical screen layout.

[0031] Fig. 4 is an illustration of the use of the bandwidth conservation scheme of the subject invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

546 a' [0032] The video surveillance system of the subject invention is specifically adapted for distributing digitized camera video on a real-time or near real-time basis over a LAN and/or a WAN. The system uses a plurality of video cameras C1, C2...Cn, disposed around a facility to view scenes of interest. Each camera captures the desired scene, digitizes the resulting video signal at a dedicated encoder E1, E2...En, respectively, compresses the digitized video signal at the respective compressor processor P1, P2...Pn, and sends the resulting compressed digital video stream to a multicast address router R. One or more display stations D1, D2...Dn may thereupon view the captured video via the intervening network N. The network may be hardwired or wireless, or a combination, and may either a Local Area Network (LAN) or a Wide Area Network (WAN), or both.

[0033] The preferred digital encoders E1, E2...En produce industry-standard MPEG-1 digital video streams. The use of MPEG-1 streams is advantageous due to the low cost of the encoder hardware, and to the ubiquity of software MPEG-1 players. However, difficulties arise from the fact that the MPEG-1 format was designed primarily to support playback of recorded video from a video CD, rather than to support streaming of 'live' sources such as cameras.

[0034] MPEG-1 system streams contain multiplexed elementary bit streams containing compressed video and audio. Since the retrieval of video and audio data from the storage medium (or network) tends to be temporally discontinuous, it is necessary to embed certain timing information in the respective video and audio elementary streams. In the MPEG-1 standard, these consist of Presentation Timestamps (PTS) and, optionally, Decoding Timestamps (DTS).

[0035] On desktop computers, it is common practice to play MPEG-1 video and audio using a proprietary software package such as, by way of example, the Microsoft Windows Media Player. This software program may be run as a standalone application, otherwise components of the player may be embedded within other software applications.

[0036] Media Player, like MPEG-1 itself, is inherently file-oriented and does support playback of continuous sources such as cameras via a network. Before Media Player begins to play back a received video file, it must first be informed of certain parameters including file name and file

length. This is incompatible with the concept of a continuous streaming source, which may not have a filename and which has no definable file length.

[0037] Moreover, the time stamping mechanism used by Media Player is fundamentally incompatible with the time stamping scheme standardized by the MPEG-1 standard. MPEG-1 calls out a time stamping mechanism which is based on a continuously incrementing 94 kHz clock located within the encoder. Moreover, the MPEG-1 standard assumes no Beginning-of-File marker, since it is intended to produce a continuous stream. In the present invention, a common MPEG-1 encoder IC is used to perform the actual MPEG compression of a digitized camera signal. The IC selected is a W99200F IC, produced by Winbond Corporation of Taiwan. This IC produces an MPEG Video Elementary Stream that contains the appropriate PTS information as mandated by the MPEG standard.

0038 2. When invoking Media Player to view the streaming camera video, it is first necessary to inform Media Player of the file length. Since the camera produces a stream rather than a discrete file, the file length is undefined. In order to overcome this problem all of the Media Player's 63-bit file length variables are set to 1. Media Player compares this value to a free-running counter that counts ticks of a 10 MHz clock. This counter is normally initialized to zero at the beginning of the file. Given 63 bits, this permits a maximum file length of approximately thirty thousand years, longer than the useful life of the product or, presumably, it's users. This effectively allows the system to play streaming sources.

[0039] The next problem arises when additional users attempt to connect to a stream that is already in progress. Media Player expects that file length and other related information is normally sent only once, in a file header, and is not periodically repeated. Thus, users connecting later will not receive the file length information contained in the header. The subject invention has overcome this problem by developing a software 'front-end' filter that examines and modifies data being passed from the network to Media Player. This software formulates a dummy video file header, and passes it to Media Player. The filter then examines the incoming video stream, finds the next sequential Video Header, and thereupon begins passing the networked video data to the Media Player decoder and renderer. This effectively allows users to 'tune in late', by providing Media Player with an appropriate file header.

[0040] A further problem arises when the networked video data is passed to Media Player. Since the user has connected to the video stream after the start of the file, the first time stamp received

by Media Player will be non-zero, which causes an error. To overcome this problem, the front-end software filter replaces each received timestamp with a value of (current time stamp minus first time stamp received), which effectively re-numbers the timestamp in the local video stream starting with an initial value of zero.

546 93
[0041] Any given source of encoded video may be viewed by more than one client. This could hypothetically be accomplished by sending each recipient a unique copy of the video stream. However, this approach is tremendously wasteful of network bandwidth. A superior approach is to transmit one copy of the stream to multiple recipients, via Multicast Routing. This approach is commonly used on the Internet, and is the subject of various Internet Standards (RFC's). In essence, a video source sends its' video stream to a Multicast Group Address, which exists as a port on a Multicast-Enabled network router or switch. The router or switch then forwards the stream only to IP addresses that have known recipients. Furthermore, if the router or switch can determine that multiple recipients are located on one specific network path or path segment, the router or switch sends only one copy of the stream to that path.

[0042] From a client's point of view, the client need only connect to a particular Multicast Group Address to receive the stream. A range of IP addresses has been reserved for this purpose; essentially all IP addresses from 224.0.0.0 to 239.255.255.255 have been defined as Multicast Group Addresses.

[0043] Unfortunately, there is not currently a standardized mechanism to dynamically assign these Multicast Group Addresses, in a way that is known to be globally unique. This differs from the ordinary Class A, B, or C IP address classes. In these classes, a regulatory agency assigns groups of IP addresses to organizations upon request, and guarantees that these addresses are globally unique. Once assigned this group of IP addresses, a network administrator may allocate these addresses to individual hosts, either statically or dynamically DHCP or equivalent network protocols. This is not true of Multicast Group Addresses; they are not assigned by any centralized body and their usage is therefore not guaranteed to be globally unique.

[0044] Each encoder must possess two unique IP addresses – the unique Multicast Address used by the encoder to transmit the video stream, and the ordinary Class A, B, or C address used for more mundane purposes. It is thus necessary to provide a means to associate the two addresses, for any given encoder.

[0045] The subject invention includes a mechanism for associating the two addresses. This

method establishes a sequential transaction between the requesting client and the desired encoder. An illustration of this technique is shown in Fig. 2.

[0046] First, the client requesting the video stream identifies the IP address of the desired encoder. This is normally done via graphical methods, described more fully below. Once the encoder's IP address is known, the client obtains a small file from an associated server, using FTP, TFTP or other appropriate file transfer protocol over TCP/IP. The file, as received by the requesting client, contains various operating parameters of the encoder including frame rate, UDP bitrate, image size, and most importantly, the Multicast Group Address associated with the encoder's IP address. The client then launches an instance of Media Player, initializes the previously described front end filter, and directs Media Player to receive the desired video stream from the defined Multicast Group Address.

[0047] First, the client requesting the video stream identifies the IP address of the desired encoder. This is normally done via graphical methods, described more fully below. Once the encoder's IP address is known, the client obtains a small file from an associated server, using FTP, TFTP or other appropriate file transfer protocol over TCP/IP. The file, as received by the requesting client, contains various operating parameters of the encoder including frame rate, UDP bitrate, image size, and most importantly, the Multicast Group Address associated with the encoder's IP address. The client then launches an instance of Media Player, initializes the previously described front end filter, and directs Media Player to receive the desired video stream from the defined Multicast Group Address.

[0048] Streaming video produced by the various encoders is transported over a generic IP network to one or more users. User workstations contain one or more ordinary PC's, each with an associated video monitor. The user interface is provided by an HTML application within an industry-standard browser, specifically Microsoft Internet Explorer.

[0049] One aspect of the invention is the intuitive and user-friendly method for selecting cameras to view. The breadth of capability of this feature is shown in Fig. 3. The main user interface screen provides the user with a map of the facility, which is overlaid with camera-shaped icons depicting location and direction of the various cameras and encoders. This main user interface has, additionally, a section of the screen dedicated to displaying video from the selected cameras.

[0050] The video display area of the main user interface may be arranged to display a single video image, or may be subdivided by the user into arrays of 4, 9, or 16 smaller video display

- Select the number of video images to be displayed within the video display area. This is done by pointing and clicking on icons representing screens with the desired number of images.
- Display a desired camera within a desired ‘pane’ in the video display area. This is done by pointing to the desired area on the map, then ‘dragging’ the camera icon to the desired pane.
- Edit various operating parameters of the encoders. This is done by pointing to the desired camera, the right-clicking the mouse. The user interface then drops a dynamically generated menu list that allows the user to adjust the desired encoder parameters.

```
// this function responds to a dragStart event on a camera
function cameraDragStart(i)
{
event.dataTransfer.setData("text",currSite.siteMaps[currSite.currMap]
.hotSpots[i].camera.id);
dragSpot = currSite.siteMaps[currSite.currMap].hotSpots[i];
event.dataTransfer.dropEffect = "copy";
dragging = true;
event.cancelBubble = true;
}
// this function responds to a dragStart event on a cell
// we might be dragging a hotSpot or a zone
function cellDragStart(i)
{
}
}
// this function responds to a drop event on a cell input element
function drop(i)
{
if (dragSpot != null) // dragging a
hotSpot
{
}
else if (dragZone != null) // dragging a
zone object
{
currMonitor.zones[i] = dragZone; // set the cell zone
dragZone = null; // null
dragZone
zoneVideo(currMonitor.id, i); // start the
video
}
else
{
}
else
{
dropCameraId(currMonitor,d,i); // setup hotSpot
startMonitorVideo(currMonitor, i); // start the
video
displayCells(); //
redisplay the monitor cells
}
```

```

}
dragging = false;
event.cancelBubble = true;
}

```

In the foregoing code, the function:

```

event.dataTransfer.setData("text",currSite.siteMaps[currSite.currMap].hotspots
[i].camera.id)

```

retrieves the IP address of the encoder that the user has clicked. The subsequent function startMonitorVideo(currMonitor, i) passes the IP address of the selected encoder to an ActiveX control that then decodes and renders video from the selected source.

[0051]

It is often the case that the user may wish to observe more than 16 cameras, as heretofore discussed. To support this, the system allows the use of additional PC's and monitors. The additional PC's and monitors operate under the control of the main user application. These secondary screens do not have the facility map as does the main user interface. Instead, these secondary screens use the entire screen area to display selected camera video.

[0052]

These secondary screens would ordinarily be controlled with their own keyboards and mice. Since it is undesirable to clutter the user's workspace with multiple mice, these secondary PC's and monitors operate entirely under the control of the main user interface. To support this, a series of button icons are displayed on the main user interface, labeled, for example, PRIMARY, 2,3, and 4. The video display area of the primary monitor then displays the video that will be displayed on the selected monitor. The primary PC, then, may control the displays on the secondary monitors. For example, a user may click on the '2' button, which then causes the primary PC to control monitor number two. When this is done, the primary PC's video display area also represents what will be displayed on monitor number two. The user may then select any desired camera from the map, and drag it to a selected pane in the video display area. When this is done, the selected camera video will appear in the selected pane on screen number

[0053]

2. Streaming video signals tend to be bandwidth-intensive. The subject invention provides a method for maximizing the use of available bandwidth by incorporating multiple resolution transmission and display capabilities. Since each monitor is capable of displaying up to 16 separate video images, the bandwidth requirements of the system can potentially be

~~enormous. It is thus desirable to minimize the bandwidth requirements of the system.~~

[0054] To address this, each encoder is equipped with at least two MPEG-1 encoders. When the encoder is initialized, these two encoders are programmed to encode the same camera source into two distinct streams: one low-resolution low-bitrate stream, and one higher-resolution, higher-bitrate stream.

Sub 25
[0055] ~~When the user has configured the video display area to display a single image, that image is obtained from the desired encoder using the higher-resolution, higher-bitrate stream. The same is true when the user subdivides the video display area into a 2 x 2 array; the selected images are obtained from the high-resolution, high-bitrate streams from the selected encoders. The network bandwidth requirements for the 2 x 2 display array are four times the bandwidth requirements for the single image, but this is still an acceptably small usage of the network bandwidth.~~

0056
[0056] However, when the user subdivides a video display area into a 3 x 3 array, the demand on network bandwidth is 9 times higher than in the single-display example. And when the user subdivides the video display area into a 4 x 4 array, the network bandwidth requirement is 16 x that of a single display. To prevent network congestion, video images in a 3 x 3 or 4 x 4 array are obtained from the low-resolution, low-speed stream of the desired encoder. Ultimately, no image resolution is lost in these cases, since the actual displayed video size decreases as the screen is subdivided. If a higher-resolution image were sent by the encoder, the image would be decimated anyway in order to fit it within the available screen area.

[0057] While specific features and embodiments of the invention have been described in detail herein, it will be understood that the invention includes all of the enhancements and modifications within the scope and spirit of the following claims.